I CLAIM:

1. A computer program product comprising a computer program operable to control a computer to detect a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said computer program comprising:

resource data reading logic operable to read resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

resource data comparing logic operable to compare said resource data with characteristics of resource data of said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

2. A computer program product as claimed in claim 1, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

3. A computer program product as claimed in claim 1, wherein said resource data comparing logic is operable to compare said resource data with characteristics of a plurality of known computer programs to detect if said packed computer program contains one of said plurality of known computer programs.

4. A computer program product as claimed in claim 1, wherein said resource data comparing logic is operable to processes said resource data of said packed computer file to generate fingerprint data and to compare said fingerprint data with fingerprint data of said known computer program.

5. A computer program product as claimed in claim 1, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

6. A computer program product as claimed in claim 1, wherein said resource data specifies for each resource item a storage location of said resource item.

7. A computer program product as claimed in claim 6, wherein said storage location of said resource item is specified as an relative offset value.

8. A computer program product as claimed in claim 1, wherein said resource data specifies for each resource item a size of said resource item.

9. A computer program product as claimed in claim 4, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

10. A computer program product as claimed in claim 4, wherein said fingerprint data includes a number of program resource items specified within said resource data.

11. A computer program product as claimed in claim 4, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

12. A computer program product as claimed in claim 4, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

13. A computer program product as claimed in claim 4, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

14. A computer program product as claimed in claim 9, wherein said checksum value is rotated between each item being added into said checksum.

15. A computer program product as claimed in claim 1, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

16. A computer program product as claimed in claim 1, wherein said packed computer file is a Win32 PE file.

17. A computer program product comprising a computer program operable to control a computer to generate data for detecting a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said computer program comprising:

resource data reading logic operable to read resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

characteristic data generating logic operable to generate characteristic data associated with said resource data for comparison with characteristics of resource data of

said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

18. A computer program product as claimed in claim 17, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

19. A computer program product as claimed in claim 17, wherein said characteristic data generating logic is operable to generate characteristic data from a plurality of known computer programs to enable detection of any of said plurality of known computer programs within said packed computer program..

20. A computer program product as claimed in claim 17, wherein said characteristic data generating logic is operable to processes said resource data of said packed computer file to generate fingerprint data for comparison with fingerprint data generated from packed computer files.

21. A computer program product as claimed in claim 17, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

22. A computer program product as claimed in claim 17, wherein said resource data specifies for each resource item a storage location of said resource item.

23. A computer program product as claimed in claim 22, wherein said storage location of said resource item is specified as an relative offset value.

24. A computer program product as claimed in claim 17, wherein said resource data specifies for each resource item a size of said resource item.

25. A computer program product as claimed in claim 20, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

26. A computer program product as claimed in claim 20, wherein said fingerprint data includes a number of program resource items specified within said resource data.

27. A computer program product as claimed in claim 20, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

28. A computer program product as claimed in claim 20, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

29. A computer program product as claimed in claim 20, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

30. A computer program product as claimed in claim 25, wherein said checksum value is rotated between each item being added into said checksum.

31. A computer program product as claimed in claim 17, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

32. A computer program product as claimed in claim 17, wherein said packed computer file is a Win32 PE file.

33. A method of controlling a computer to detect a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said method comprising the steps of:

reading resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

comparing said resource data with characteristics of resource data of said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

34. A method as claimed in claim 33, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

35. A method as claimed in claim 33, wherein said step of comparing compares said resource data with characteristics of a plurality of known computer programs to detect if said packed computer program contains one of said plurality of known computer programs.

36. A method as claimed in claim 33, comprising processing said resource data of said packed computer file to generate fingerprint data and comparing said fingerprint data with fingerprint data of said known computer program.

37. A method as claimed in claim 33, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

38. A method as claimed in claim 33, wherein said resource data specifies for each resource item a storage location of said resource item.

39. A method as claimed in claim 38, wherein said storage location of said resource item is specified as an relative offset value.

40. A method as claimed in claim 33, wherein said resource data specifies for each resource item a size of said resource item.

41. A method as claimed in claim 36, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

42. A method as claimed in claim 36, wherein said fingerprint data includes a number of program resource items specified within said resource data.

43. A method as claimed in claim 36, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

44. A method as claimed in claim 36, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

45. A method as claimed in claim 36, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

46. A method as claimed in claim 41, wherein said checksum value is rotated between each item being added into said checksum.

47. A method as claimed in claim 33, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

48. A method as claimed in claim 33, wherein said packed computer file is a Win32 PE file.

49. A method of controlling a computer to generate data for detecting a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said method comprising the steps of:

reading resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

generating characteristic data associated with said resource data for comparison with characteristics of resource data of said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

50. A method as claimed in claim 49, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

51. A method as claimed in claim 49, wherein said step of generating generates characteristic data from a plurality of known computer programs to enable detection of any of said plurality of known computer programs within said packed computer program..

52. A method as claimed in claim 49, comprising processing said resource data of said packed computer file to generate fingerprint data for comparison with fingerprint data generated from packed computer files.

53. A method as claimed in claim 49, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

54. A method as claimed in claim 49, wherein said resource data specifies for each resource item a storage location of said resource item.

55. A method as claimed in claim 54, wherein said storage location of said resource item is specified as an relative offset value.

56. A method as claimed in claim 49, wherein said resource data specifies for each resource item a size of said resource item.

57. A method as claimed in claim 52, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

58. A method as claimed in claim 52, wherein said fingerprint data includes a number of program resource items specified within said resource data.

59. A method as claimed in claim 52, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

60. A method as claimed in claim 52, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

61. A method as claimed in claim 52, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

62. A method as claimed in claim 57, wherein said checksum value is rotated between each item being added into said checksum.

63. A method as claimed in claim 49, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

64. A method as claimed in claim 49, wherein said packed computer file is a Win32 PE file.

65. Apparatus for detecting a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said apparatus comprising:

a resource data reader operable to read resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

a resource data comparitor operable to compare said resource data with characteristics of resource data of said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

66. Apparatus as claimed in claim 65, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

67. Apparatus as claimed in claim 65, wherein said resource data comparitor is operable to compare said resource data with characteristics of a plurality of known computer programs to detect if said packed computer program contains one of said plurality of known computer programs.

68. Apparatus as claimed in claim 65, wherein said resource data comparitor is operable to processes said resource data of said packed computer file to generate fingerprint data and to compare said fingerprint data with fingerprint data of said known computer program.

69. Apparatus as claimed in claim 65, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

70. Apparatus as claimed in claim 65, wherein said resource data specifies for each resource item a storage location of said resource item.

71. Apparatus as claimed in claim 70, wherein said storage location of said resource item is specified as an relative offset value.

72. Apparatus as claimed in claim 65, wherein said resource data specifies for each resource item a size of said resource item.

73. Apparatus as claimed in claim 68, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

74. Apparatus as claimed in claim 68, wherein said fingerprint data includes a number of program resource items specified within said resource data.

75. Apparatus as claimed in claim 68, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

76. Apparatus as claimed in claim 68, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

77. Apparatus as claimed in claim 68, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

78. Apparatus as claimed in claim 73, wherein said checksum value is rotated between each item being added into said checksum.

79. Apparatus as claimed in claim 65, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

80. Apparatus as claimed in claim 65, wherein said packed computer file is a Win32 PE file.

81. Apparatus for generating data for detecting a known computer program within a packed computer file, said packed computer file being unpacked upon execution, said apparatus comprising:

a resource data reader operable to read resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and being readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file; and

a characteristic data generator operable to generate characteristic data associated with said resource data for comparison with characteristics of resource data of said known computer program to detect a match with said known computer program indicative of said packed computer file containing said known computer program.

82. Apparatus as claimed in claim 81, wherein said known computer program is one of:

a Trojan computer program; and

a worm computer program.

83. Apparatus as claimed in claim 81, wherein said characteristic data generator is operable to generate characteristic data from a plurality of known computer programs to enable detection of any of said plurality of known computer programs within said packed computer program..

84. Apparatus as claimed in claim 81, wherein said characteristic data generator is operable to processes said resource data of said packed computer file to generate fingerprint data for comparison with fingerprint data generated from packed computer files.

85. Apparatus as claimed in claim 81, wherein said program resource items used by said known computer program include one or more of:

icon data;

string data;

dialog data;

bitmap data;

menu data; and

language data.

86. Apparatus as claimed in claim 81, wherein said resource data specifies for each resource item a storage location of said resource item.

87. Apparatus as claimed in claim 86, wherein said storage location of said resource item is specified as an relative offset value.

88. Apparatus as claimed in claim 81, wherein said resource data specifies for each resource item a size of said resource item.

89. Apparatus as claimed in claim 84, wherein said fingerprint data includes a checksum value calculated in dependence upon one or more of:

a number of program resource items specified beneath each node within hierarchically arranged resource data;

string names associated with program resource items within said resource data; and

sizes of program resource items within said resource data.

90. Apparatus as claimed in claim 84, wherein said fingerprint data includes a number of program resource items specified within said resource data.

91. Apparatus as claimed in claim 84, wherein said fingerprint data includes a location within said resource data of an entry specifying a program resource item having a largest size.

92. Apparatus as claimed in claim 84, wherein said fingerprint data includes timestamp data indicative of a time of compilation of said known computer program.

93. Apparatus as claimed in claim 84, wherein said fingerprint data includes a flag indicating which data is included within said fingerprint data.

94. Apparatus as claimed in claim 89, wherein said checksum value is rotated between each item being added into said checksum.

95. Apparatus as claimed in claim 81, wherein said packed computer file includes an unpacking computer program which upon execution decompresses said known computer program.

96. Apparatus as claimed in claim 81, wherein said packed computer file is a Win32 PE file.